MODULE:3 (React - Components, State, Props)

1. What is React Js?

ANS: The React.js framework is an open-source JavaScript framework and library developed by Facebook. It's used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

In React, you develop your applications by creating reusable components that you can think of as independent Lego blocks. These components are individual pieces of a final interface, which, when assembled, form the application's entire user interface.

React's primary role in an application is to handle the view layer of that application just like the V in a model-view-controller (MVC) pattern by providing the best and most efficient rendering execution. Rather than dealing with the whole user interface as a single unit, React.js encourages developers to separate these complex UIs into individual reusable components that form the building blocks of the whole UI. In doing so, the ReactJS framework combines the speed and efficiency of JavaScript with a more efficient method of manipulating the DOM to render web pages faster and create highly dynamic and responsive web applications.

2. What is NPM in React Js?

ANS: NPM is short for node package manager, an onli

ne directory that contains the various already registered open-source packages. NPM modules consume the various functions as a third-party package when installed into an app using the NPM command npm install .

3. What is Role of Node Js in react Js?

ANS: NodeJS is a framework of JavaScript which is mainly used for working with the backend of our application or building the backend using JavaScript, whereas ReactJS is a JavaScript front-end library. It is mainly used for building the user interface or the frontend of our application.

4. What is CLI command In React Js?

ANS: React CLI (aka rct) helps you to automate the most obvious tasks which we as developers used to do manually such as creating a new component by making a separate folder & then making component's jsx or css files manually. rct helps you out with these kind of tasks by providing you simple to use command line tool.

5. What is Components in React Js?

ANS: Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. Components come in two types, Class components and Function components, in this tutorial we will concentrate on Function components.

6. What is Header and Content Components in React Js?

ANS: React Bootstrap 5 Headers component

Headers are compositions that extend standard navbar functionalities. They contain additional components like a jumbotron, sub-navbar, or image covers which serve as a containers for extra navigation elements - usually links, forms, or call-to-action buttons.

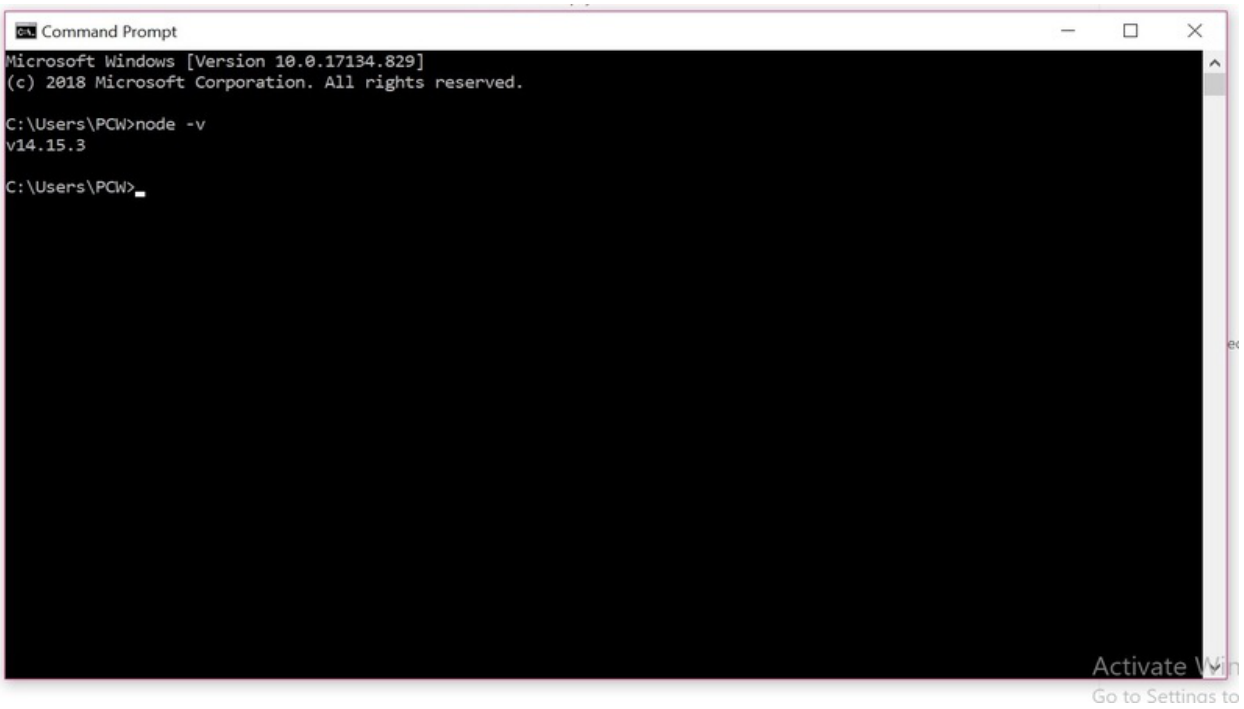
7.How to install React Js on Windows, Linux Operating System? How toinstall NPM and How to check version of NPM?

ANS: *Installation Reactjs on Windows:*

Step 1: Install Node.js installer for windows. Click on this link. Here install the LTS version (the one present on the left). Once downloaded open NodeJS without disturbing other settings, click on the **Next** button until it's completely installed.

Step 2: Open command prompt to check whether it is completely installed or not type the command –>

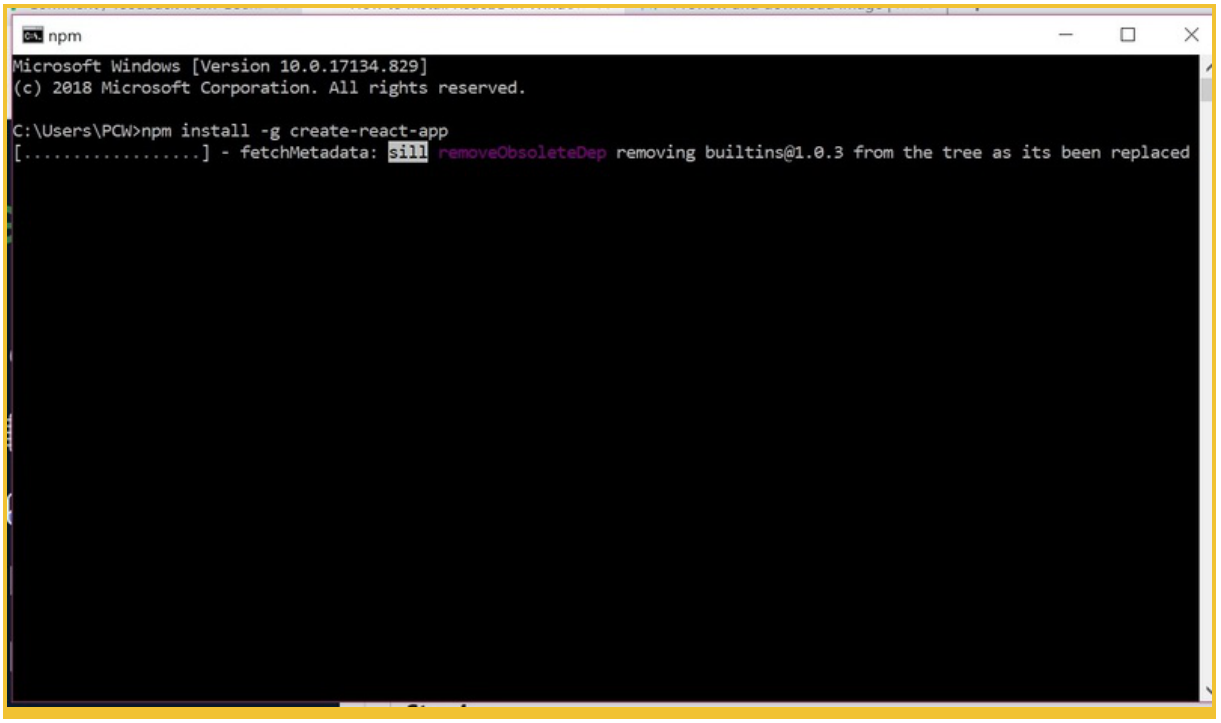
```
node -v
```



If the installation went well it will give you the version you have installed

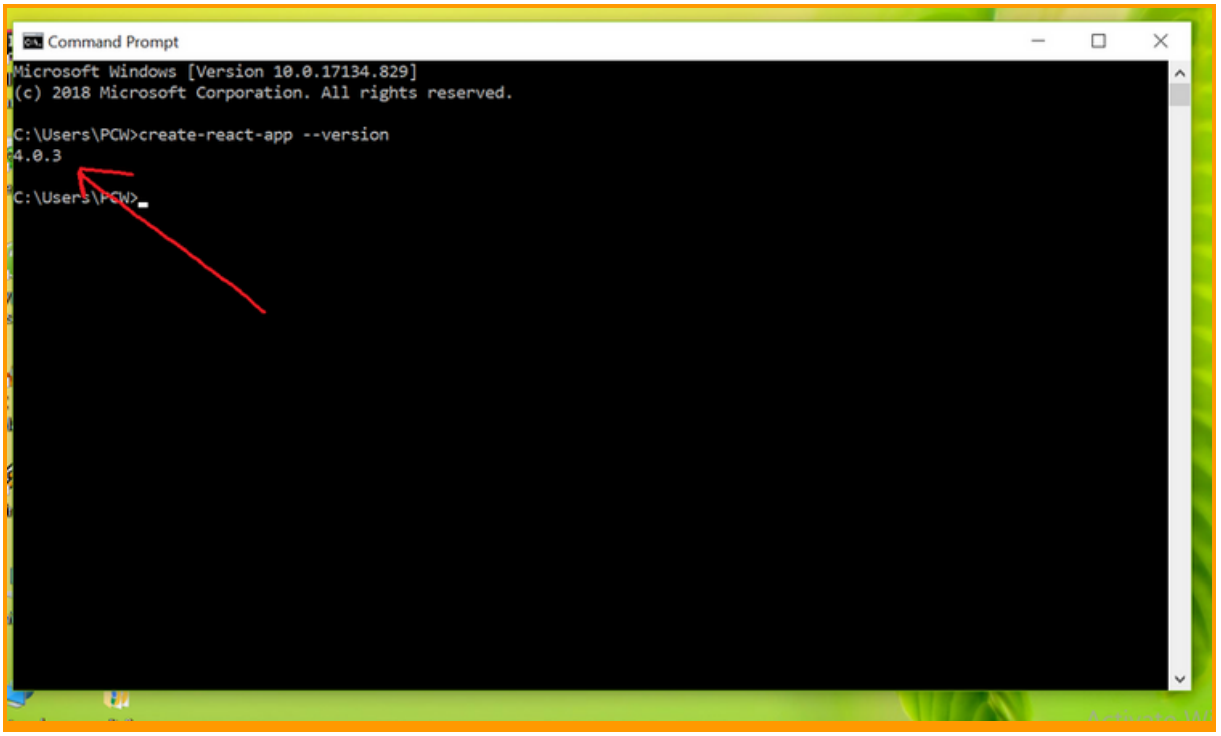
Step 3: Now in the terminal run the below command:

```
npm install -g create-react-app
```



It will globally install react app for you. To check everything went well run the command

```
create-react-app --version
```



If everything went well it will give you the installed version of react app

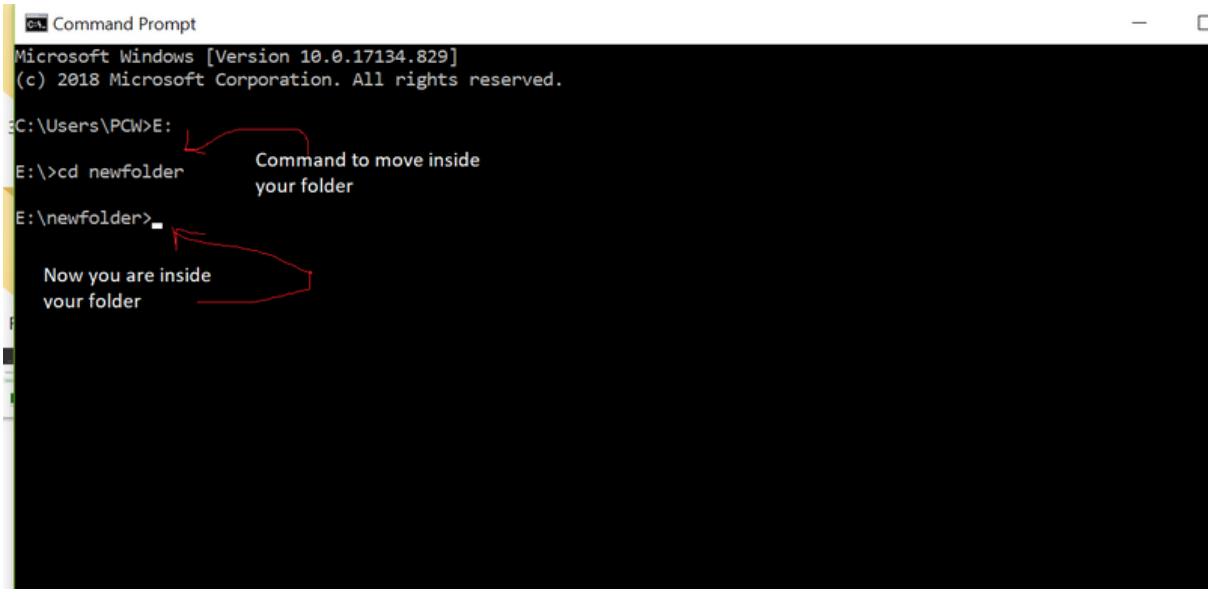
Step 4: Now Create a new folder where you want to make your react app using the below command:

```
mkdir newfolder
```

Note: The *newfolder* in the above command is the name of the folder and can be anything.

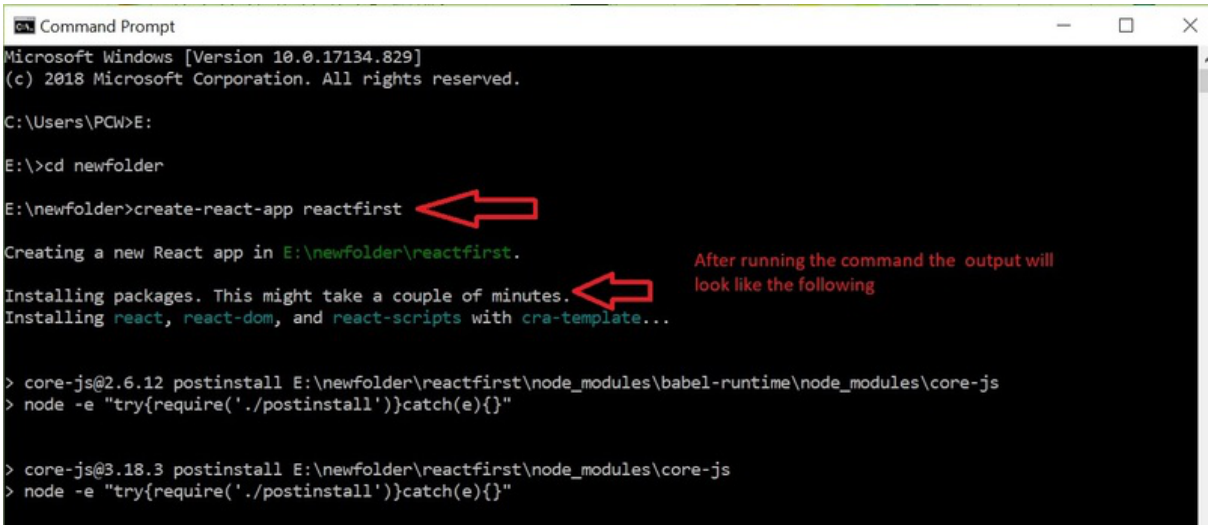
Move inside the same folder using the below command:

```
cd newfolder (your folder name)
```



Step 5: Now inside this folder run the command →

```
create-react-app reactfirst YOUR_APP_NAME
```



It will take some time to install the required dependencies

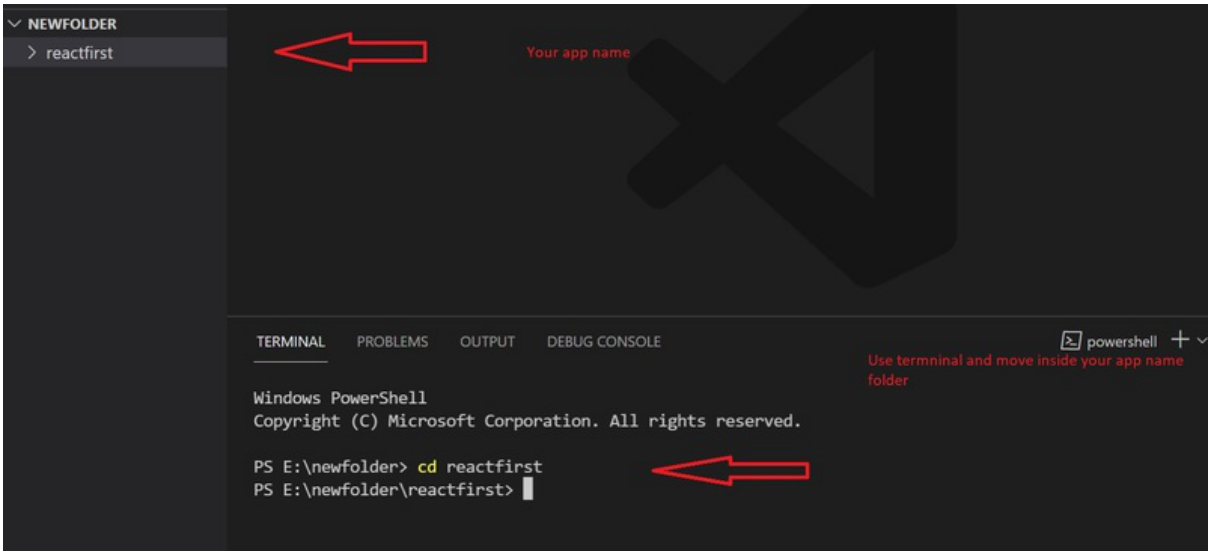
NOTE: Due to npm naming restrictions, names can no longer contain capital letters, thus type your app's name in lowercase.

```
C:\Users\sa552>create-react-app FirstReact
Cannot create a project named "FirstReact" because of npm naming restrictions:

  * name can no longer contain capital letters

Please choose a different project name.
```

Step 6: Now open the IDE of your choice for eg. Visual studio code and open the folder where you have installed the react app **newfolder** (in the above example) inside the folder you will see your app's name **reactapp** (In our example). Use the

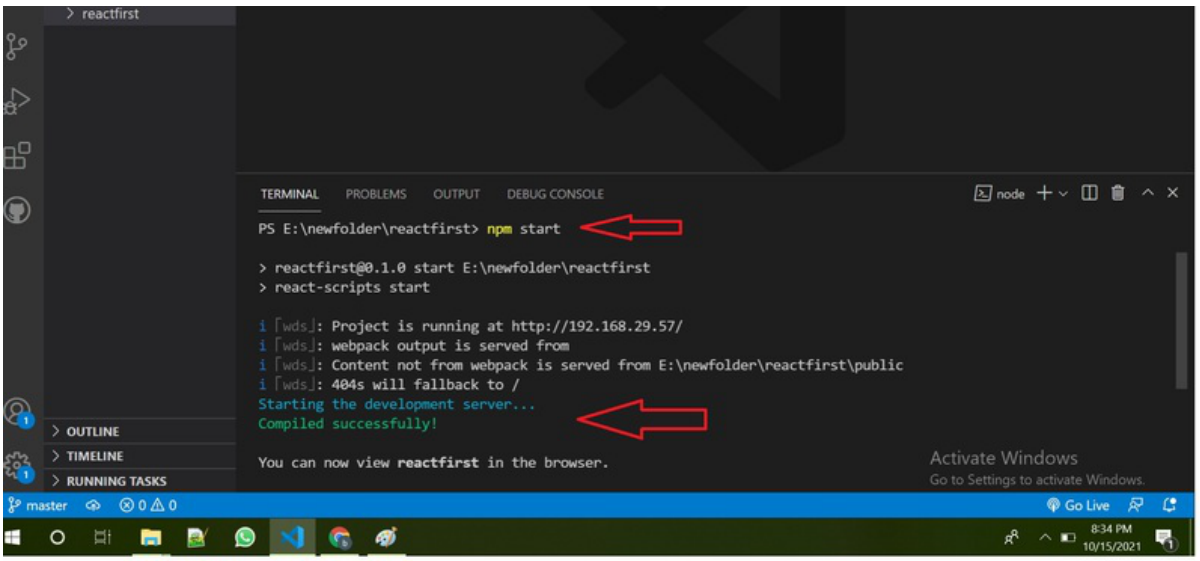
terminal and move inside your app name folder.Use command

cd reactapp (your app name)

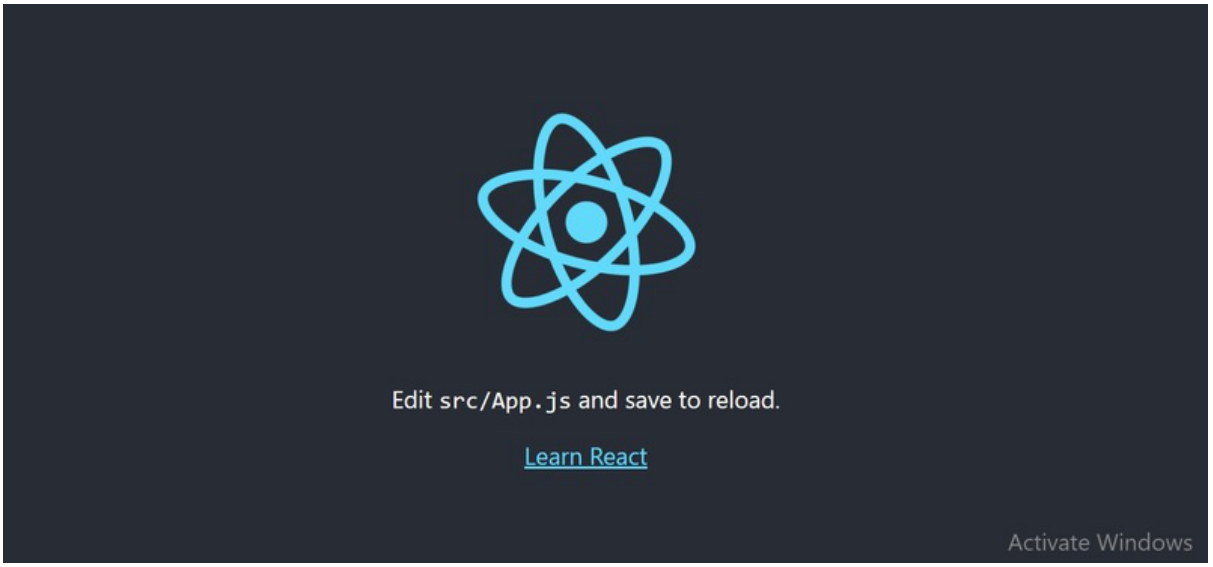


Step 7: To start your app run the below command :

```
npm start
```



Once you run the above command a new tab will open in your browser showing React logo as shown below :



Congratulation you have successfully installed the react-app and are ready to build awesome websites and app

8. How to check version of React Js?

ANS: To check which React version is your project using you need to open the package. json. Take a look under the dependencies section. It should list all of the dependencies of your project and one of those should be React.

```
npm view react version
```

9. How to change in components of React Js?

ANS: To change the state of the React component is useful when you are working on a single page application, it simply replaces the content of the existing component for the user without reloading the webpage.

We have to set initial state value inside constructor function and set click event handler of the element upon which click, Results in Changing state. Then pass the function to the click handler and change the state of the component inside the function using **setState**. The **setState** function used to change the state of the component directly or with the callback approach as mentioned below.

Example 1: This example illustrates how to change the state of the component on click.

index.js:

Javascript

```
import React from 'react'
import ReactDOM from 'react-dom'
import App from './App'
```

```
ReactDOM.render(<App />,
document.querySelector('#root'))
```

App.js:

```
import React, { Component } from 'react'


class App extends Component {
  constructor(props){
    super(props)


    // Set initial state
    this.state = {msg : 'Hi, There!'}


    // Binding this keyword
    this.handleClick =
this.handleClick.bind(this)
  }


  handleClick(){
```

```
    // Changing state
    this.setState({msg : 'Welcome to the React
world!'})
  }

  render(){
    return (
      <div>
        <h2>Message :</h2>



<p>{this.state.msg}</p>




        {/* Set click handler */}
        <button onClick={this.handleClick}>
          Click here!
        </button>
      </div>
    )
```

```
  }
}


export default App
```

10. How to Create a List View in React Js?

ANS: Input:

ChekBoxList.js

```
import React from 'react'

import styles from './my-style.module.css'



function ChekboxList() {

  return (

    <div>

      {/* chekbox start function */}

     <div className="col-md-5 mt-5 offset-4">
```

```
<h1>The "React way" to Reander a list</h1>

</div>

<div className="col-md-3 mt-5 offset-4">

  <div className="input-group">

    <div className="input-group-text">

      <input className="form-check-input mt-0"
type="checkbox" defaultValue aria-label="Checkbox for following
text input" />

    </div>

    <input type="text" className="form-control"
aria-label="Text input with checkbox" />

  </div>

</div>

<div className="col-md-3 mt-5 offset-4">

  <div className="input-group">

    <div className="input-group-text">
```

```
          <input className="form-check-input mt-0"
type="checkbox" defaultValue aria-label="Checkbox for following
text input" />
        </div>
        <input type="text" className="form-control"
aria-label="Text input with checkbox" />
       </div>
      </div>
      <div className="col-md-3 mt-5 offset-4">
       <div className="input-group">
        <div className="input-group-text">
          <input className="form-check-input mt-0"
type="checkbox" defaultValue aria-label="Checkbox for following
text input" />
        </div>
        <input type="text" className="form-control"
aria-label="Text input with checkbox" />
```

```
        </div>
      </div>
      <div className="col-md-3 mt-5 offset-4">
        <div className="input-group">
          <div className="input-group-text">
            <input className="form-check-input mt-0"
type="checkbox" defaultValue aria-label="Checkbox for following
text input" />
          </div>
          <input type="text" className="form-control"
aria-label="Text input with checkbox" />
          {/* chekbox end function */}
        </div>
      </div>
    </div>
  )
```

```
}



export default ChekboxList
```

my-style.module.css

```
div{
    background-color: rgb(115, 222, 115);
}
```

App.js

```
import React from "react";
import ChekBoxList from "./ChekBoxList"



function App() {
  return (
      <ChekBoxList />
  );
}
```

export default App;

Output:

11. Create Increment decrement state change by button click?

Ans. App.js

```
import React, {useState} from "react";

function App() {

  //Increment decrement function

  const [count, setCount] = useState(0);

  return (

    <>


   <div className="container my-5">

    <div className="card text-center my-5">
```

```
      <div className="card-body">
        <h1>Counter app</h1>
        <div className="my-5">
          <h2 className="my-5">{count}</h2>
          {/* button start */}
          <button className="btn btn-success mx-3" onClick={() =>
setCount(count + 1)}>Increment</button>
          <button className="btn btn-danger mx-3" onClick={() =>
setCount(count - 1)} disabled={count === 0}>Decrement</button>
          <button className="btn btn-secondary mx-3" onClick={()
=> setCount(0)} disabled={count === 0}>Reset</button>
          {/* button end */}
        </div>
      </div>
    </div>
  </div>
```

```
    </>

  );

}

export default App;
```

Output.